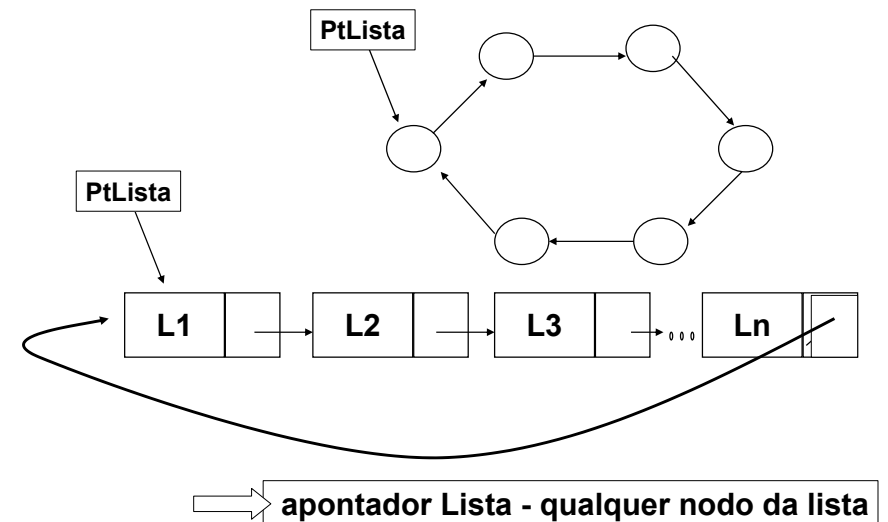


Lista Circular

Lista encadeada circular

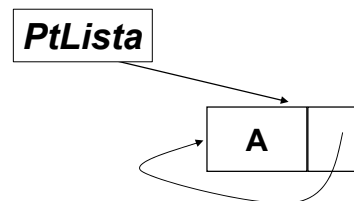


Lista encadeada circular

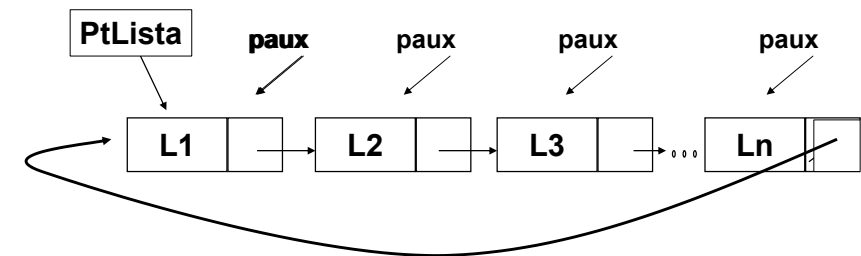
- lista vazia

PtLista = nil

- lista com 1 só nodo



Percorrer lista encadeada circular



- parar quando o nodo inicial for novamente alcançado

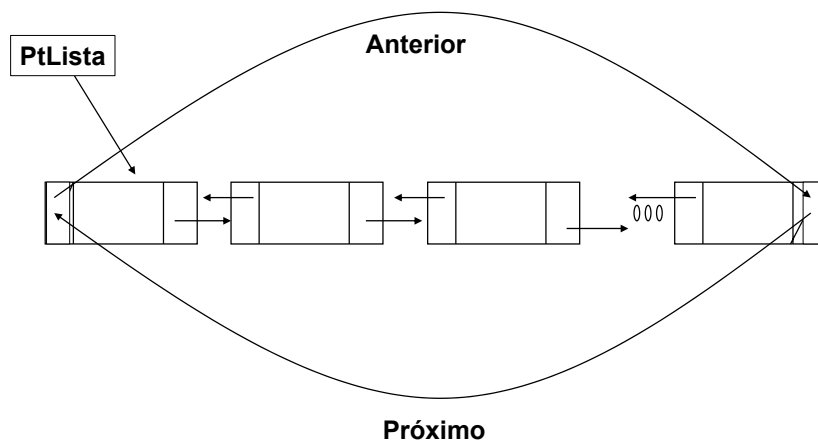
Imprimir dados de lista circular encadeada

```
Proc ImprimirLLCircular (var PtLista: TipoPtnodo);  
  var paux: TipoPtnodo;  
início  
  se PtLista = nil  
  então escrever ( 'Lista vazia ! ' )  
  senão início  
    paux := PtLista;      { paux vai percorrer a lista }  
    repita  
      escrever ( paux↑.info );    { operação solicitada }  
      paux := paux↑.prox; { avança para o próximo }  
    até que ????? = ?????;  
    fim;  
fim ImprimirLLCircular;
```

Imprimir dados de lista circular encadeada

```
Proc ImprimirLLCircular (var PtLista: TipoPtnodo);  
  var paux: TipoPtnodo;  
início  
  se PtLista = nil  
  então escrever ( 'Lista vazia ! ' )  
  senão início  
    paux := PtLista;      { paux vai percorrer a lista }  
    repita  
      escrever ( paux↑.info );    { operação solicitada }  
      paux := paux↑.prox; { avança para o próximo }  
    até que paux = PtLista; { para quando volta ao início }  
    fim;  
fim ImprimirLLCircular;
```

Lista circular duplamente encadeada



Algoritmos sugeridos:

- contruir lista circular duplamente encadeada, lendo dados do teclado
- inserção de novo nodo em LL circular simplesmente encadeada, logo após um nodo que contém um determinado campo de informação
- mesmo anterior, no caso de duplamente encadeada
- remoção de um nodo que contém uma determinada informação, nos dois casos (simplesmente e duplamente encadeada)
- destruição de lista circular